THE RESOURCE SHARER

DIGITOGRAPH
P. O. Box 4249
Tucson, Arizona 85717

DEC 1974 RSTS-11 SPECIAL INTEREST GROUP NEWSLETTER VOL.1 NO. 3

FROM THE EDITOR

THOSE OF YOU WHO WERE UNABLE TO ATTEND THE FALL DECUS SYMPOSIUM MISSED A VERY INFORMATIVE AND HELPFUL MEETING. THE STATE AND FUTURE OF RSTS WHERE OUTLINED BY GERRY HORNIK AND CREW(SEE LATER ARTICLES).

IN ADDITION THE SIG HAD A VERY LIVELY AND CONSTRUCTIVE SESSION. SEVERAL THINGS NECESSARY FOR THIS TO BECOME AN EFFECTIVE SIG WHERE DISCUSSED. THESE INCLUDED:

1. MORE TIMELY ISSUES OF THE NEWSLETTER-
THE PROBLEMS THAT OCCURED IN GETTING ISSUES 1 & 2 MAILED WERE DISCUSSED. MS. BARBARA KOWALCZYK, THE DECUS SIG COORDINATOR,EXPLAINED THE REASONS FOR THE LONG DELAYS AND PROMISED TO MAKE SURE THAT FUTURE ISSUES WOULD BE MAILED OUT WITHIN 2 WEEKS OF MY SENDING THE ORIGINAL NEWSLETTER TO HER. I'M MAILING THIS ISSUE TO HER ON NOV. 26, SO WE WILL SEE IF SHE CAN DO IT.

2. NEWSLETTER NOT REACHING ALL SIG MEMBERS-
MANY OF THE SIG MEMBERS HAVE NOT BEEN RECEIVING THE NEWSLETTER. BARBARA HAS PROMISED TO ME A LIST OF ALL THE SIG MEMBERS AND FROM NOW ON I WILL PRINT MAILING LABELS TO INSURE THAT EACH MEMBER GETS HIS NEWSLETTER. ONE PROBLEM THAT HAS OCCURED IS THAT USER'S HAVE JOINED THE SIG WITHOUT JOINING DECUS, WHICH IS AGAINST THE DECUS BY-LAWS.

3. THE NEED FOR FORMALIZED RSTS STANDARDS-
TO FACILITATE THE EXCHANGE OF PROGRAMS BETWEEN USERS, MANY OF THE SIG MEMBERS FELT THE NEED TO DEVELOP FILE STRUCTURE STANDARDS AND BASIC PLUS PROGRAMMING STANDARDS. ROSS MILLER HAS VOLUNTEERED HIS SHOP'S FILE STRUCTURE STANDARDS AND MERRILL LIKES HAS VOLUNTEERED HIS SHOP'S BASIC-PLUS PROGRAMMING STANDARDS TO SERVE AS STARTING POINTS FOR THESE VENTURES. COPIES OF BOTH APPEAR LATER IN THIS NEWSLETTER. WE ARE PLANNING SIG WORKSHOPS FOR THE SPRING DECUS MEETING TO HAMMER OUT THESE STANDARDS. IF YOU ARE INTERESTED IN HELPING PLEASE GET IN TOUCH WITH ROSS OR MERRILL.

4. PARTICIPATION IN THE SIG-
I NEED YOUR HELP....! SEND ME YOUR GOOD IDEAS FOR RSTS AND THE SIG. SEND ME INFORMATION ON THAT NEAT LITTLE SYSTEM PROGRAM YOU HAVE WRITTEN OR THE STRANGE UNSOLVED BUG YOU HAVE FOUND IN RSTS OR SOMETHING THE SIG NEEDS TO BE WORKING ON. THUS FAR ONLY 6 PEOPLE HAVE SENT ME ANY ARTICLES FOR THE NEWSLETTER. I HOPE TO GET A NEWSLETTER OUT MONTHLY, BUT I CAN'T WRITE THEM ALONE SITTING HERE IN DURANT, OKLAHOMA. SEND THEM TO:

BOB BRANTON
COMPUTER CENTER
SOSU
DURANT,OKLAHOMA 74701
(405) 924-0121 EXT 395

RSTS/E V5B DISCUSSION AT DECUS

DURING THE DECUS RSTS SESSION THE NEW FEATURES OF RSTS/E V5B WERE DISCUSSED. THE FOLLOWING ARE THE HIGHLIGHTS OF THAT DISCUSSION AS I PERCEIVED THEM.

A. SYSTEM GENERATION FEATURES.

- .WILL COME ON MAGTAPE, DECTAPE, OR DECPACK.
- .GENERATION IS OPTIMIZED BY DEVICE TYPE.
- .SEVERAL GENERATION CHECKPOINTS HAVE BEEN ESTABLISHED. THIS SHOULD AID WITH GENERATION ERRORS & MODIFICATIONS.
- .PREVISION FOR AUTO-ANSWERS OF HARDWARE CONFIGURATION QUESTIONS.

B. NEW SYSTEM FACILITIES.

- .DOCUMENT PREPARATION SYSTEM- RUNOFF. THIS 14K PROGRAM PROVIDES ABOUT 75% OF THE FEATURES OF THE PDP-10 VERSION.
- .BATCH CAPABILITY FOR BASIC PROGRAMS. THE BATCH COMMANDS ARE COMPATIBLE WITH RSX-11D BATCH.
- .LINE PRINTER SPOOLING- INCLUDES THE ABILITY TO RECOVER FROM JAMS AND PRINTER OFF-LINE ERRORS DURING A 10 MINUTE GRACE PERIOD, SO IF YOUR QUICK YOU WON'T BE FLUSHED.
- .2780 COMMUNICATION CAPABLIITY UNDER RSTS- OPTION(ONLY? $4000).

C. NEW PROGRAMMING FEATURES.

- .INTERJOB COMMUNICATION(SEND/RECEIVE) BY JOB NAME OR #.
- .PSEUDO KEYBOARDS
- .MULTI-TERMINAL JOBS.
- .SCALED ARITHMETIC AS PART OF 4 WORD REAL MATH PACKAGE. ALLOWS PRECISE CALCULATION WITH USER DEFINED SCALES. (IE. DECIMAL TYPE ARITHMETIC)
- .BINARY UNFORMATED TERMINAL I/O.
- .NON-ASCII CARD READING(IE. BINARY CARD READING).
- .REASSIGNMENT OF DEVICES TO OTHER JOBS.
- .REATTACHMENT OF JOBS TO TERMINALS WITHOUT LOGIN.
- .ACCOUNTING DATA AVAILABLE 1 WEEK WITHOUT POSSIBLE OVERFLOW.

D. EASIER PROGRAMMING FEATURES.

- .STING$- TO GENERATE STRINGS OF REPEATED CHARACTERS.
- .CVT$$ FUNCTION.
- .FILENAME SCANNING WITH "WILD CARDS".
- .DIRECTORY LOOKUP BY FILENAME USING "WILD CARDS".
- .PRE-EXTENTION OF FILES WHEN OPENED USING "FILESIZE" CLAUSE.
- .EXAMINATION OF BUFFERSIZE & CHARACTERISTICS ON OPEN CHANNELS.

E. NEW UTILITIES.

- .MOUNT/DISMOUNT OF USER DISKS BY USERS (UMOUNT).
- .QUICK COPY OF TAPES & DISKS (COPY).
- .QUICK FLEXIBLE DIRECTORY LISTING (DIRECT).
- .CROSS REFERENCE LISTING (CREF).
- .PROGRAM NUMBER RESEQUENCE (RESEQ).
- .ON-LINE INITIALIZATION OF PRE-FORMATED DISKS.
- .RETRIVEAL OF FILES FROM DOS/BATCH DISK FILES(RK & RP).

F. DOCUMENTATION.

THIS IS WHAT SEEMS TO HAVE HELD UP THE RELEASE OF V5B SO LONG. THE STORY IS THAT THE CODE WAS FROZEN JULY 22, BUT THE RELEASE HAS BEEN ON HOLD WAITING FOR THE COMPLETION OF 10 MANUALS COMPRISING 1500 PAGES. THE MANUALS INCLUDE: BASIC LANGUAGE MANUAL, SYSTEM USER'S GUIDE,SYSTEM MANAGER'S GUIDE,RUNOFF USER'S GUIDE, DOS/BATCH TEXT EDITOR MANUAL,SYSTEM RELIABILITY TEST MANUAL,INSTALLATION NOTES, 2780 USER'S GUIDE,SORT USER'S GUIDE, AND COMMERCIAL USER'S GUIDE.

G. SPECIAL FEATURES.

HIDDEN OVERLAP SEEKS FOR MOVING HEAD DISKS. THIS FEATURE WHILE NOT PUBLICIZED CAN CAUSE MAJOR SYSTEM SPEED IMPROVEMENTS. IT MAY BE INVOKED AT SYSTEM GENERATION TIME BY ANSWERING THE NUMBER OF MOVING HEAD DISKS QUESTION WITH "0,*DISKS".

MULTIPLE SWAP FILES. SWAP FILES MAY BE PLACED OVER EVERAL DISKS.

SETUP OF ONE JOB AT SYSTEM STARTUP TO FIX TERMINAL CHARACTERISTICS FOR REMOTE LINES(NO MORE CONFIG). IT SEEMS TO ME THAT THIS COULD CAUSE SOME PROBLEMS TO THOSE OF YOU THAT HAVE TERMINALS OF SEVERAL DIFFERENT SPEEDS SHARING THE SAME LINE.ONCE A USER SPEEDS THE LINE UP IT WILL REMAIN SET AT THAT SPEED. A PROBLEM COULD OCCUR FOR EXAMPLE IF A USER SET THE LINE TO 300 BAUD AND LATER ANOTHER USER TRIED TO USE THE LINE WITH A TELETYPE. THE SECOND USER WOULD HAVE NO WAY TO SLOW THE LINE DOWN TO 110 BAUD.

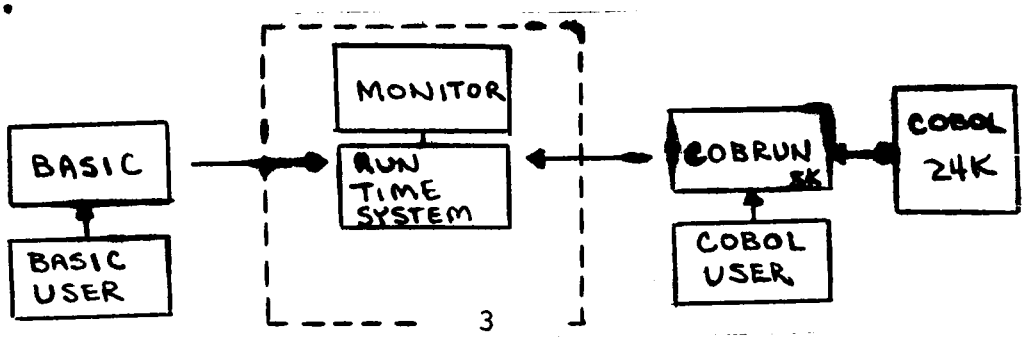
****

RSTS/E = WHERE DOES IT GO FROM HERE

PART OF THE DECUS RSTS SESSION WAS SPENT DISCUSSING DEC'S FUTURE PLANS FOR RSTS/E. DENNEY PAVLOCK,RSTS SOFTWARE DEVELOPMENT MANAGER, DISCRIBED THE ON-GOING DEVELOPMENT. HE STATED THAT THE RSTS PROGRAM STAFF BEEN INCREASED FOR 3 MEMBERS TO 13 MEMBERS. PRESENTLY 6 PEOPLE ARE WORKING ON MAINTAINING RSTS AND THE REST ARE DEVELOPMENT PEOPLE.

THE MAJOR GOAL OF THE V5C RELEASE OF RSTS/E IS TO PROVIDE SUPPORT FOR THE NEW PERIPHERAL DEVICES FOR THE 11. THIS INCLUDES THE RJP04 DISK DRIVES AND THE TU16 MAGTAPE DRIVE. NO FIRM RELEASE DATE FOR V5C WAS SET, BUT IT LOOKS LIKE MID-YEAR 1975 IS A REASONABLE ESTIMATE.

FUTURE FUTURES.....!!!! HAVE YOU BEEN WISHING FOR THAT SECOND LANGUAGE RUNNING UNDER RSTS? WELL THERE MAY BE SOME DAYLIGHT AT THE END OF THE TUNNEL. TALK HAS IT THAT MAYBE IN A YEAR COBOL(RSX-11/D VARIETY) WILL RUN UNDER THE RSTS MONITOR. IT PROBABLY WILL BE TIED TO RSTS/E V5C VIA A SEPARATE RUNTIME MONITOR(COBRUN). THE MONITOR WILL REQUIRE ABOUT 5K AND COBOL WILL REQUIRE ABOUT 24K OF MEMORY. IT MAY ONLY BE SINGLE USER DEPENDING ON THE LOAD EFFECT IT HAS ON RSTS. IT DEFINITELY WILL NOT BE FOR THE USER WITH SMALL SOFTWARE BUDGETS. COBOL WILL COST $7000. BELOW IS A DIAGRAM OF THE SYSTEM .



RSTS V4 - THE DEATH OF A SYSTEM

REMEMBER RSTS WITHOUT THE E FROM BACK IN THE NICE YEARS WHEN INFLATION WAS ONLY 7%. IT USE TO BE SOLD BY DIGITAL EQUIPMENT CORPORATION. WELL 5 MINUTES OF THE DECUS RSTS SESSION WAS SPENT DISCUSSING V4.

IF YOU ARE STILL ONE OF WHOSE UNCOUNTED RSTS V4 USERS THEN COME FEBRARY OR MARCH YOU WILL RECEIVE YOUR LAST RSTS CHRISTMAS PRESENT FROM DEC. IT WILL BE CALLED V4B AND WITH ANY LUCK IT WILL BE SENT TO YOU FREE OF CHARGE. IT WILL HAVE A FEW NICE FEATURES LIKE A BETTER SCHEDULER ,PRIORITY SYS-CALLS, AND MANY OF THE RSTS/E CUSPS. IT SHOULD CLEAR UP THE BUGS IN V4A. BUT YOU HAD BE BETTER BE QUICK IN FINDING ANY BUGS IN V4B, BECAUSE COME AMERICA'S BICENTENNIAL YEAR AND RSTS V4 SPR SUPPORT WILL SAG AND RSTS V4 WILL GO THE WAY OF LATIN.

****

SPR'S AND RSTS SOFTWARE SUPPORT

A GREAT DEAL OF TIME DURING THE RSTS SESSIONS AT DECUS WAS SPENT DISCUSSING SPR'S. MANY USERS EXPRESSED CONCERN OVER THE LONG RESPONSE TIMES THEY HAVE BEEN RECEIVING ON THEIR SPR'S AND THE EVEN LONGER TIMES IT TAKES TO GET FIXES DISTRIBUTED TO ALL USERS. DENNIE PAVLOCK , RSTS SOFTWARE DEVELOPMENT MANAGER, STATED THAT HIS PEOPLE HAD SPEEDED UP THE SPR TURN-AROUND TIME AND ARE ATTEMPING TO PROVIDE A CONSISTANTLY FAST RESPONSE. HE ALSO SAID THAT HE WOULD LOOK INTO THE POSSIBLITY OF PRINTING THE RAW SPR'S IN THE SOFTWARE DISPATCH. IT WAS HIS CONTENTION THAT PRINTING TIME HAS BECOME THE MAJOR HOLDUP IN THE "SOFTWARE DISPATCH".

IF YOU HAVE A RAW SPR OR AN SPR WITH A PATCH THAT YOU FEEL NEEDS IMMEDIATE DISTRIBUTION TO USERS, SEND ME A COPY AND I'LL PUT IT IN THE NEWSLETTER. THIS IDEA HAS WORKED VERY EFFECTIVELY FOR THE RSX-11 SIG.

****

WISHES

NOW THAT DEC IS CHARGING $900/YEAR FOR "SOFTWARE MAINTANCE SERVICES" , IT SEEMS TO ME THAT USERS SHOULD HAVE MORE VOICE IN THE DESIGN OF FUTURE RELEASES OF RSTS/E. TO OFTEN THE FEATURES OF A NEW RSTS RELEASE HAVE BEEN BORN FROM A NEED SOME DEC PROGRAMMING WIZARD HAD DURING A CRASH PROJECT ON WHICH HE WAS USING RSTS. SINCE THE RSTS USERS ARE NOW PAYING FOR ON-GOING DEVELOPMENT, THEY SHOULD NOW HAVE AT LEAST A FORMAL METHOD TO SUGGEST THEIR DESIRES FOR RSTS EXTENTIONS.

I PURPOSE THAT DEC DEVELOP A "WISH" REQUEST FORM SIMILAR TO THE SPR FORM. ON THESE "WISH" FORMS USERS COULD STATE THEIR DESIRED RSTS FEATURES. DEC IN TURN WOULD IN RETURN ANSWER THE "WISH" REQUEST WITH AN INDICATION OF ITS PHYSICAL FEASIBILITY AND WHAT PRIORITY AND TIME FRAME(IF ANY) THE RSTS DEVELOPMENT PEOPLE WOULD ATTACH TO THE PROPOSED FEATURE .

ROBERT BRANTON

IF YOU FEEL AS I DO ABOUT THE "WISH" WRITE DEC THRU:

DENNIS PAVLOCK
RSTS DEVELOPMENT MANAGER
DEC
146 MAIN ST.
MAYNARD,MASS. 01754

(PS- SEND ME A COPY OF YOUR LETTER)

NOTICES

THIS IS A NEW FEATURE OF THE NEWSLETTER DESIGNED TO GIVE SIG MEMBERS AN OPPORTUNITY TO REQUEST INFORMATION FROM OTHER USERS AND TO DESCRIBE AVAILABLE PROGRAMS OR SERVICES. IF YOU HAVE SOMETHING FOR THIS COLUMN MAKE IT AS SHORT AS POSSIBLE AND SEND IT TO BOB BRANTON.

******WANTED******

LOOKING FOR OTHER USERS WORKING IN THE AREA OF LARGE HEAVY MANUFACTURING USING RSTS SYSTEMS. IF YOU ARE INTERESTED IN EXCHANGING INFORMATION WRITE:

DAVID W. JOZA
WYMAN-GORDON CO.
HARVEY,ILL. 60246
(312) 333-1400

******NOTICE******

ANY RSTS EDUCATIONAL USERS, WHO HAVE NOT JOINED EDUSIG, NOW IS YOUR CHANCE. IF YOU WISH TO JOIN OR HAVE AN ARTICLE FOR THE EDUSIG NEWSLETTER WRITE:

ALICE D. PETERS
DEC EDUCATIONAL PRODUCTS
146 MAIN ST.
MAYNARD,MASS 01754

******WANTED******

LOOKING FOR A RSTS LINEAR PROGRAMMING PACKAGE. NEED LINEAR PROGRAMMING MODULES, A MATRIX GENERATOR, AND A REPORT WRITER.

BILL WILKIE
CROWN ZELLERBACH CO.
#1 BUSH
SAN FRANCISCO,CA 94119

******FOR SALE******

RSTS V5 LANGUAGE MANUALS(MAY 73)- CLEARAGE IN MULTIPLES OF 10 @ $6.00 PLUS SHIPPING.

COMPUTER CENTER
NORTHFIELD MOUNT HERMON SCH
P.O. BOX 86
MT. HERMON,MASS 01354

******WANTED******

LOOKING FOR ANYONE WHO HAS INTERFACTED TU60'S WITH RSTS/E V5A.

JOHN F. HASSE
6701 AUTUMN VIEW COURT
SYKESVILLE,MD 21784

******WANTED******

1. JUNIOR HIGH OR HIGH SCHOOL TUTORING PACKAGE(S)
2. TEST SCORING PROGRAMS FOR MARK SENSE CARDS
3. INFORMATION ON WRITING AND READING PDP11 DECTAPES ON A PDP8
4. NDSL & NSL LOAN ACCOUNTING UNDER RSTS

RICHARD LEE
DATA PROCESSING DEPT
AUGUSTANA COLLEGE
SIOUX FALLS,S.D. 57102

******WANTED*****

LOOKING FOR OTHER RSTS USERS IN THE MEDICAL FIELD INTERESTED IN FORMING A SIG.

BRYAN WEYRICK
MEDICAL ARTS LAB.
1111 N. LEE #254
OKLAHOMA CITY,OK 73103

PROGRAM FOR EXCHANGE

Title: Time Shared General Accounting System

Author: Young Door Computer Services Staff

Abstract: The complete generalized package provides systems for Accounts Receivable and Sales Analysis, Accounts Payable and General Ledger, Payroll, Consolidating Financial Statements, and various utility routines. All of the programs are documented, as are the files, complete with operating instructions, sample reports, and typical program dialogue. The multi-terminal system is completely on line.

Configuration: Contact Jack Bouknight, Data Processing Services Manager, Young Door Company, Midwest Division, Post

Cost: Office Box 460, Plymouth, Indiana 46563, for additional information and prices.

****

THE REMAINDER OF THIS NEWSLETTER IS DEVOTED TO RSTS STANDARDS. ROSS MILLER AND MERRILL LIKES HAVE KINDLY ALLOWED ME TO PRINT THE STANDARDS LETTERS THAT THEY SENT ME EARLIER IN THE YEAR. WE HOPE THAT THESE CAN SERVE AS THE STARTING POINT FOR SOME FILE AND BASIC PROGRAMMING STANDARDS THAT WILL FACILITATE USER EXCHANGE OF PROGRAMS. HAVING STANDARDS WILL NOT WORK UNLESS THE SIG MEMBERS GET INVOLVED, SO HELP ROSS AND MERRILL OUT. WRITE THEM AND LET THEM KNOW YOUR IDEAS FOR STANDARDS AND VOLUNTEER TO SERVE WITH THEM ON THE SIG STANDARDS COMMITTEE.

information technology, inc.

February 16, 1974

Mr. Robert Branton
Southeastern State College
Computer Center
Durant, Okla.

Mr. Branton:

On the attached sheet are a list of the most common programming conventions used at INFOTECH. I would appreciate your comments and suggestions on revision and expansion of these standards. It is my intention to propose the revised set of standards to the RSTS SIG and get their reaction to determine if it is feasible for BASIC shops to have standards to aid in the transfer of programs and data files. Your help would be appreciated.

Sincerely,

Merrill E. Likes, Jr.

Merrill E. Likes, Jr.

1) Allocation of statement number ranges for particular purposes; for example...
   a. General program code at 1-4999
   b. Subroutines unique to this program at 5000-8999
   c. ON ERROR GOTO at 9000-9999
   d. Subroutines common to various programs at 10000-19999 (this includes file record layouts)
   e. I-O subroutines at 20000-29999
   f. Subroutines available to all programs in a system at 30000-32499 (this includes XLATE string creation, date reformat,...)
   g. Program wrapup at 32500-32749
   h. END at 32750

   Items a,b,c,g would typically be user coded, while items d,e,f would be created through the APPEND feature (for V5, else via merging of code with PIP).

2) Allocation of variables for particular purposes; for example...
   a. XLATE strings of the form Tn$
   b. Loop iteration control of the form In%

3) Statement numbers aligned in positions 1-5, and first character of code at TAB position 9. Other subsets of code (such as nested IF) would be indented at succeeding TAB positions. For example...

```
    1   OPEN "FILNAM.EXT" FOR INPUT AS FILE 3%
  101   IF A0$=A1$ GOTO 32601
32601   CLOSE 3%
32750   END
or...
00010   OPEN "FILNAM.EXT" FOR INPUT AS FILE 3%
00110   IF A0$=A1$ THEN LSET H0$=A1$
        ELSE GOSUB 7500:
                IF A0$=M5$ GOTO 110 ELSE 90
32600   CLOSE 3%
32750   END
```

4) Include in program documentation a cross-reference sheet showing variable name, expanded name, and description where applicable; for example...

   a.

A0$	Account status ("I"-inactive)
A1$	Account number, six characters
A2$	Account name, 25 characters
A3$	Account address, 40 characters
An$	Account...

DIGITEC
P.O. Box 43
Tucson, Arizona 85717

5) For batch type job streams (programs CHAINed together) as might be used in a typical processing cycle, the job name and beginning and ending times are logged on the console in a set format; for example...

a.
```
00010   PRINT: PRINT "*** JOBNAM   BGN "; TIME$(0%)
.....
32740   PRINT "*** JOBNAM   END "; TIME$(0%)
```

6) Inquiry / update programs are kept in non-privileged accounts, while maintenance type programs (backup, load, merge to YTD trans) are kept in privileged accounts. This should help prevent terminal operators from accessing file structures other than what they are designated to, while allowing the console operator to perform necessary maintenance to multiple file structures as needed.

7) Separate logical sections of a program with dummy statements and descriptive headers; for example...

a.
```
07500
            !   SUBROUTINE TO CALCULATE STATE TAX

07510   T=...
```

8) For the benefit of Field Service, the system exercisers are kept in an account with proj, prog and password the same for most local installations; for example...

a.
```
#1,49
PASSWORD:DECFS   (D E C Field Service)
```

9) ?????

AMERICAN SIGN & INDICATOR CORPORATION
HOME OFFICE N.2310 FANCHER WAY, SPOKANE, WASHINGTON ● FOR MAIL: P.O. DRAWER 2727, 99220 ● PHONE (509) 535-4101

July 31, 1974

Mr. Robert Branton
Computer Center
Southeastern Oklahoma State College
Durant, OK 74701

Dear DECUS Member:

The following few pages outline a proposed Basic Plus standard for file layouts and usage. Without such a standard, it is very difficult to share useful programs and information between different RSTS users, or even within different departments in one user's company.

Also included are descriptions of three generalized programs which employ these file structure techniques to allow a development programmer to manipulate data files without having to write special programs for this purpose.

Please review this material and if you find serious oversights, please respond with a letter or call so we can adopt a standard of some type at the DECUS convention this fall in San Diego, California, at the RSTS SIG meeting.

Sincerely,

Ross Miller
Chairman, RSTS SIG Standards
Senior Programmer/Analyst

American Sign and Indicator
2310 North Fancher Way
Spokane, Washington 99220
(509)535-4101

mc/087

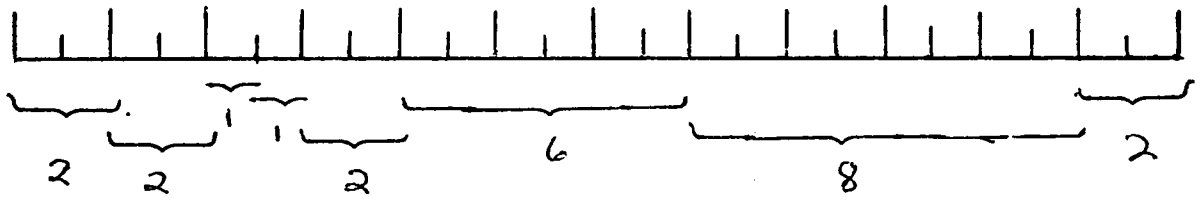
Programming Standards:

** Control Z should always move you in direction of an exit.
** Always use "input line" statement when asking for file name.
** When asking for a file either setup a default if "cr" is typed without a name
or re-ask the question.
** Minimize length of time a disk block is in lock mode.
** In all programs unless you intend to create a file use:
Open L$ for input as ----------
** Get the logical record length and header offset and all other information from header
** Don't use any " " in open statement or open for output.
Instead set varibles at beginning of program to the string value
100 L$="DP1:FOO.TMP"
.
.
.

23000 Open L$ for input as File 1%

** When setting up variable for file name use logical device name
** Line 30000 will be the entry point for 'ccl'
** Line 31000 will be the chained entry point
** Line 32000 will be entry point from non-logged in terminals
** Line 20000-21000 will be reserved for general system functions used by everyone
** Lines 25000-26000 will be error handling routines
** Lines 10000-20000 will be reserved for subroutines and local functions
** All programs will end with line 32767; thus the END statement will be at that line number
** The body of the program will lie between lines number 1 & 10000 with a minimum
seperation of 10 lines for the rough draft, thus allowing 10 lines to be inserted for later modification.

The Purpose of this Document is to Describe the Structure of the new HEADER DESCRIPTION



Number of Bytes	Byte Position	Description
2	1,2	Number of logical records in file. (If -1 See floating point number below)
2	3,4	Number of sorted logical records in file.
1	5	Number of fields per Logical Record or zero
1	6	Number of logical records used for header (1 or more)
2	7,8	Length of logical record. Logical Records never smaller than 4 characters & never larger than 512.
6	9-14	Flag word
(8)	15-22	For floating point - number of records in file
(2)	23,24	Number of records in use.

All other bytes used, if needed, to describe file. They are non-standard.

( ) parenthesis indicate optional fields.

Flag Word of Header Record

Byte 1 will contain a code describing the status of the file.

C - Cleaning	(file is being restructured for more efficient access)
L - Locked	(Don't use file closed now)
R - Read only	(Don't update or add information)
A - Active	(Everything is OK)

Byte 2 will be the code to tell operator when a new binary key is needed.

B - File is okay
S - File needs to be resorted

Bytes 3, 4, or 5 are left open for programmer to decide what he needs for a particular file.

Byte 6 - This is to be coded starting with first letter of alphabet.

For example:
During clean process (which is run with a batch file) each program after it has run successfully puts a letter in Byte 6. If batch bombs out the programer can look at the file and see what the number was. If necessary more than one byte may be used for this coding.

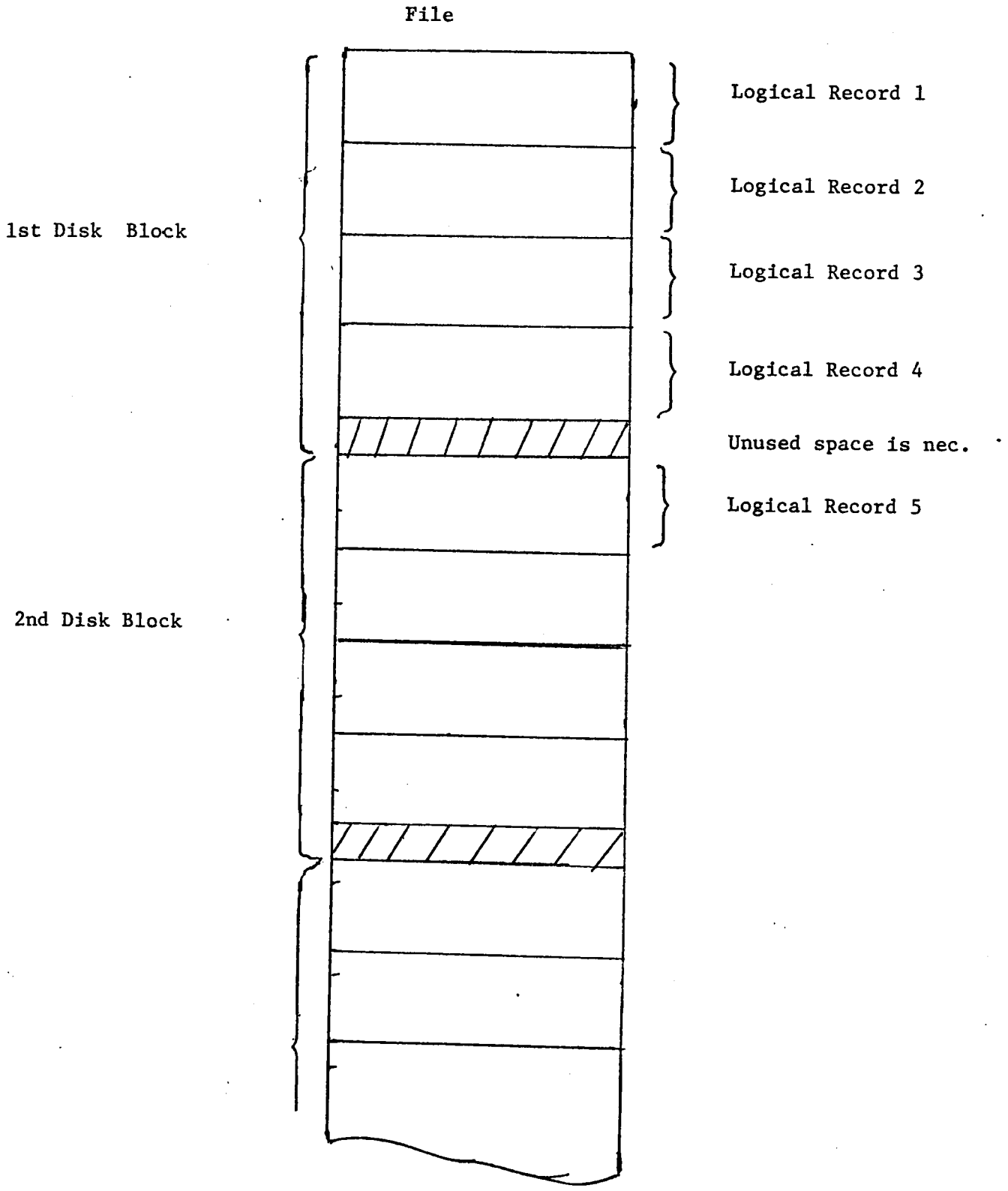


Diagram showing the relationship between Physical Disk Blocks and Logical Records.
(Note Logical Records never overlap Physical Disk Blocks)

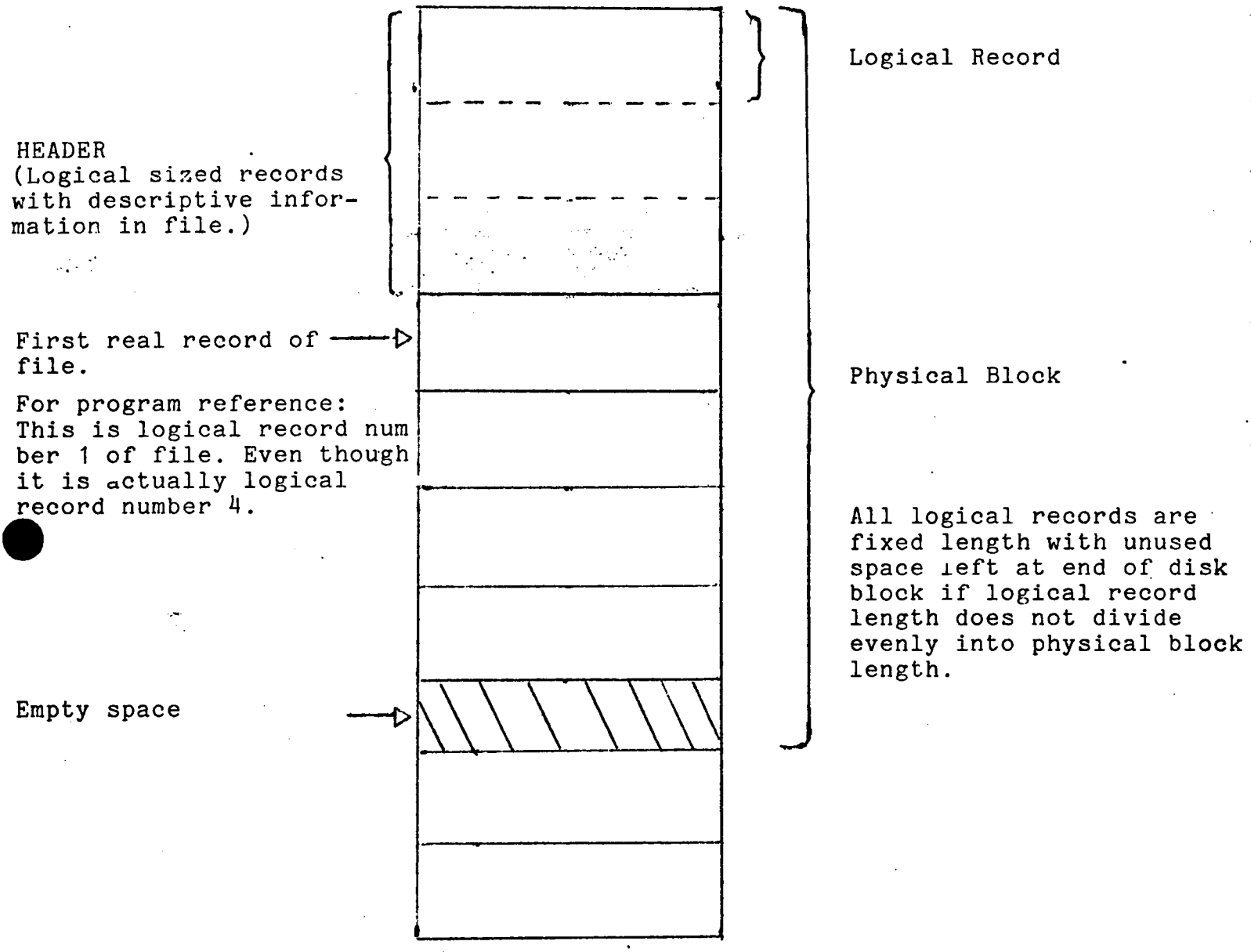
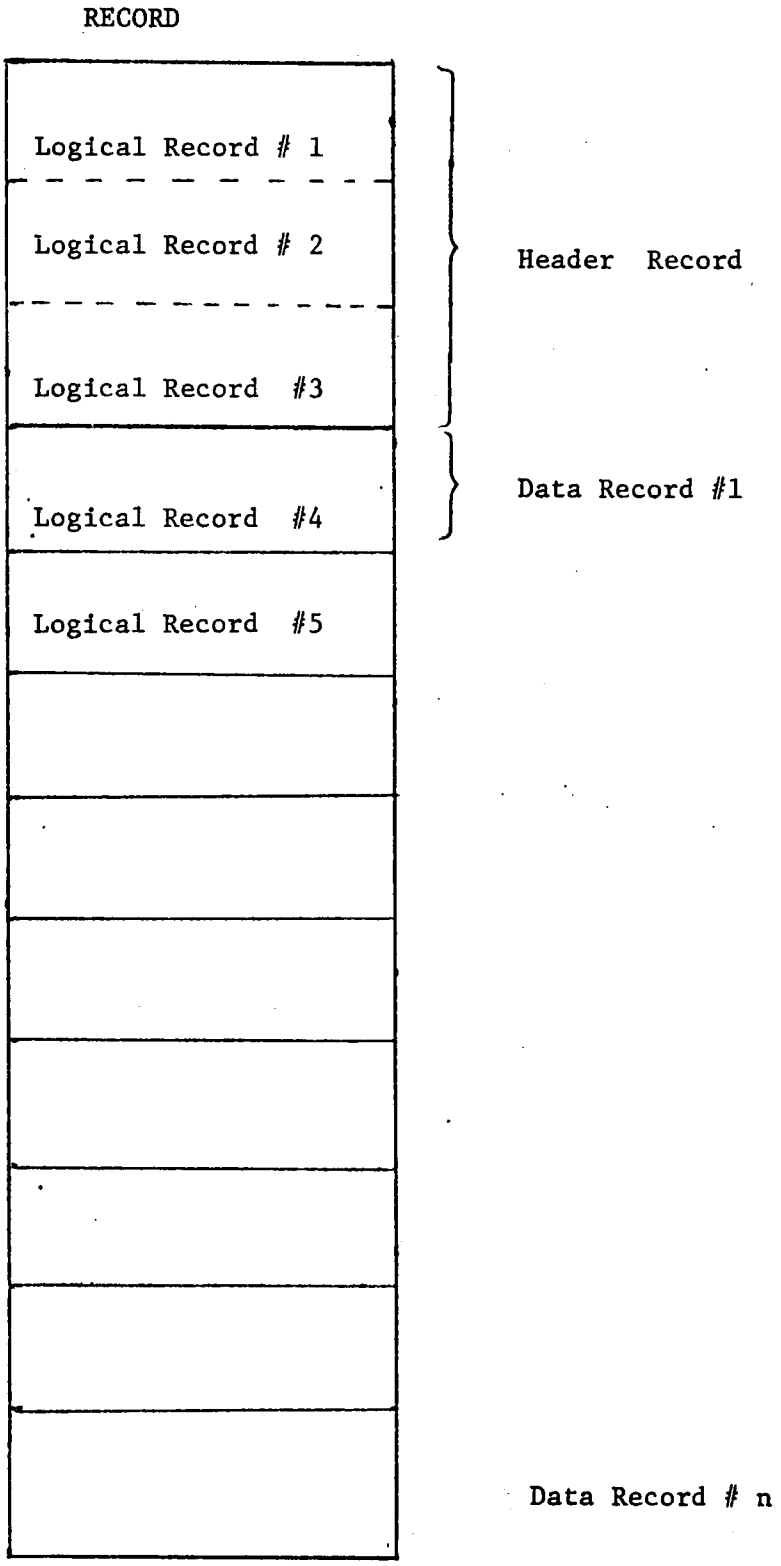

Logical Record
HEADER
(Logical sized records with descriptive information in file.)
First real record of file.
Physical Block
For program reference:
This is logical record number 1 of file. Even though it is actually logical record number 4.
All logical records are fixed length with unused space left at end of disk block if logical record length does not divide evenly into physical block length.
Empty space

Diagram showing the relationship between Header Records, Data Records, and Physical Disk Blocks. (Note both Header Records and Data Records are Logical Records of equal size.)



Data file if thought of only in terms of Logical Record and forgetting about Disk Blocks.

Table B-1
Key File Header Record Contents For Sort Package

Data Type*	Core Common	Length** (in Bytes)	Description
%	P% (0%)	2	Total number of key records in the file.
%	P% (11%)	2	Number of sorted key records in the file.
%	K%	1	Number of key fields per data record.
%	H%	1	Header offset (equal to the number of key records allocated to the header).***
%	P% (10%)	2	Key record length (in bytes) including the index.
%	B%	2	The type data file. If B% is less than zero, it indicates that the data file is type 2. If B% equals 1, a type 1 data file with a header record is indicated. If B% is greater than 1, it indicates that the data file is type 1 without a header record.
%	P% (5%)	2	Length (in bytes) of the logical data record.
$		23	Full file specification of the data file (src).
%.		.4 up to 60	Key type specification, two words per key field. The first word contains the starting byte position in the logical data record plus 1000 times the key field type code. (See Appendix E for values of the type code.) The second word is the length (in bytes) of the key field.
		39 min 95 max Length	

* The percent character (%) indicates integer type (CVT%$ format) and the dollar sign ($) indicates ASCII string type.
** The total length of the header record is from 39 to 95 bytes rounded up to the next logical key record boundary.
***Header offset of 8 = first good log. Rec is in Rec 8 when starting count with 0.

Record I\O Blocking & Deblocking of Logical Records

In the example below of how to handle blocking and deblocking of Logical Records using the Basic-Plus Record I/O Features the following symbols are defined.

P2% The current block number in the buffer which is open on channel #2.

B% The logical data record that I need.

L% The length of the logical record for channel #2.

O% The number of logical records used for the header on channel #2.

To implement function call

```
.
.
.
100 R% = FNR%(B%,L%,O%) If R% <_P2% then Put
      #2%, Record P2% : Get #2%, Record r% : P2% = R%
110 Field #2%, I9%*L% as Z$,.......(I9%*1% as Z$) is used
to skip over the correct number of Logical Records in
this block
.
.
.
2000 DEF FNR%(R%,L%,X%) : R%=R%+X% : F9%=512%/L%
      : FNR%,F8%=R%/F9%+1% : I9%=R%-(F8%-1%)*F9% :
      FNEND
.
.
.
R%= Logical record
L%= Length of logical record number wanted
X%= Header offset (number of log. rec.'s used by -1)
F9%=Number of logical records within physical block
FNR%,F8%=Physical block that record wanted is in
I9%=Number of logical records to pass over before
    logical record wanted
```

The next few pages give a breif explanation of some logical record I/O utility programs.

IODUMP

Gives an unformatted printout of what is in the record I\O file. Will begin & stop anywhere within the file. It will do selective printing of the fields within file.

User interaction:

1) File to dump?
   This is the file to be printed. There is no default. The program looks at header records in file and prints the following:

   There are __#__ records in File to Dump
   Updated records=_____Fields=_____
   Header offset=_____Logical rec. leng.=_____

Then is will ask:

2) Key file?
   <CR> means no file exists. If there is a file - it prints only number of records in key file.

3) Starting with record?
   Default is 1 if number is <1 or > what is in file.

4) Ending with record?
   Default is end of file or key file.
   Otherwise the number inputed.

5) Dump to?
   Default is KB: otherwise can be a device or disk file.

6) Step size?
   Default is one. Used to spotcheck.
   Will skip the number of record inputted.

7) Record length?
Record length of file to dump. Must be between 3 & 512.
(only asked if length in Header is bad)

8) Report Width?
Default is 72. Must be between 60 & 132.

9) Field description from?
This question is asking where to look for a description of the fields wanted on printout. Default is a file in same account user is signed in under and its name is the "file-to-dump" with an extension of .DES. This file is automatically created the first time this file is dumped. (and question (9) is input form KB:)
If a <CR> or KB: is typed then following question is asked:

Please enter the following items?
End with control/z.

Name, Type(A,D,S,B,I,J,R,T), Length

"Name" is anything wanted for a header for a field.

"Length" is how long the field is.

Type Codes For Fields Are:
A - ASCII field - (Alpha or Alphanumeric)
D - Returns Julian date on integer field
S - Skip field or any number of fields
B - Byte*
F - Floating point*
I - Integer*
J - Puts spaces between columns when printed out
T - Returns time on integer field
R - Will repeat the same code as above it.
Number of lines to repeat is in "name".
How many times to repeat is in "length".
This repeats the same format(s) not the actual field(s).

*For floating point "F" and integer "I" and byte "B", length is taken as the amount of space to allow on printout.

These codes are referring to the way each field is stored in the file.

This program allows you to edit a record I/O file.

User interaction:

1. File to edit?
   This is file you want to edit.
   If you line feed after typing in
   name of file you will have read/
   write privileges ,otherwise any
   thing else is read only.

Program will look at the header and print:

There are # records in file-to-edit
Updated Records = _____ fields=_____
Header offset=_____ log rec. len.____

Then it will ask:

2. Key File?
   <CR> means no key file.
   Will only work with number
   of records in key file.

3. Record length?
   Only asks this question .
   if header record length <
   4 or _than 512 char.

4. Field description from?
   Looking for a description
   of how each field in file
   was set up. Default is a
   file in same account as user,
   with an extension of ".DES",
   and file name same as file-to-edit.
   This file is automatically
   created the first time this file
   is dumped and question (4) is
   answered with "KB:"or <CR>

If a <CR> is typed or KB: the follwiing
questions will be asked:

Please enter the following items?

End with control /Z.
Name,type(A,D,S,B,I,J,R,T),length?

"Name" is header for field.
"Length" is length of field.

Codes are:
A-ASCII field-Alpha or alphanumberic
D-Returns Julian date on integer field
S-skip field or any # of fields
B-Byte *
F-Floating point *
I-Integer *
J- puts spaces between fields on printout
T-Returns time (integer field)
R-Will repeat codes above it.
   Number of lines to repeat is in "Name"
   How many times to repeat it is in "Length".
   This repeat "formats" not the actual field.

* For "F","B",and "I" be sure and put a length
  long enough for printout.

Control/Z when completed

Program will print:

R* or R/W* depending on read or Read/ Write privileges.

Commands to enter to edit:

D: new device
        This changes the device that the text
        is printed on.
E: Record # - Record # , Field
        Will only edit on field at a time.
Prints:

Record # , Field #, type of field, whats in field followed by a "?"
        If <CR> the field is left unchanged.
        If put in new info and <CR> the field
        is changed.Will print this line for
        every record asked for.

L: Record # - Record #, Field
        Exactly like edit except it will just
        list the fields.

C: Record # - Record # , field , new info

This change command lists same as "edit"and "list" except after prints what was there it prints new info you put in. It changes every field in every record you want changed requardless of what is in the field .

S: Record # - Record # , Field , Info
<
=
>

This command will search every record you ask for and every field until it finds what your asking for in the field.
When found it is printed in same way as "list". This is a selective list command.

R: Record # - Record # , Field ,Old info , New info
<
=
>

This command does a selective list (search) and when found it is replaced with new info.

When using the Search and Replace commands you can use more than one operator. (>,<,=)

H	prints a "help" text
L	prints a list of field names and numbers. Same as in description setup.
?	Will print Rec #, Field #, and field name . This is the last record that was referenced.
ALT	prints next field in same record.
\	prints previous field in same record.
LF	goes to next record and prints same field.
^	goes to previous record and prints same field.

Generalized Report Generator

This program will do generalized Report formatting with totals and subtotals on any fields in a Record I/O file. Also it will perform calculations on numeric fields.

User Interaction:

A) File to report on?
Any Record I/O file will work here.

B) Key file?
This is a file that has the sorted key to the file input in question (A). If you don't have a key file type <CR>
(Must be Record I/O)

C) There are "?" records in file.
This is a statement that is printed telling you the # of records in the file input in question (A).

D) Starting with record #?
This question allows you to start anywhere in your file. Is asking for a number. If the complete file is wanted then <CR>

E) Ending with record #?
Allows you to end anywhere in your file. Is asking for a specific #. If the complete file is wanted then <CR>

F) Report printed on?
If you <CR>; report will print on terminal. If you specify "LP:" or "LP1:" report will print directly to either line printer. If you want the report to go to a disk file then specify a "filename.ext" To get a report on a printer you must then RUN $ QUE; using this file name.

G) Record length?
If your file was created with cards using "CARDIO" the length will be "80" other wise it is the length given when file was created.

H) Input field description from?
If you put in a file name it will automatically be renamed to have a filename of whatever you input in question (A) with an extension of "DRP" This filename should have been an already created file to auto matically answer questions (1-4) If you type "KB:" or <CR> the follow ing questions will be asked?

1) Please enter the following items.
End with <CR>
Type (A,S,B,F,I,D,R,N), Length?
If you don't use one of the above letters you will receive this message: "Type is wrong";& give you another chance. <CR> when completed.

N- Only ASCII numeric where calculations are to be performed on field.

A- Only alpha-numeric field

I- Can not be used if input file was created with cards. File is carried as CVT.

B- Byte - CHR

F- Floating point - CVT

D- Date (I field) Julian date

S- Skip field or fields

None of the above three(B,F,D) can be used if file was created with cards.

R- You can specify arithmetic calculations on certain fields.
For example:

R,F1*F2

This would multiply field #1 by field #2 and put answer in the next field. You must specify the next field also. If only performing calcs. on one field; answer will remain there.

2) Total items?
Asking for a field # that you want a total for. If more than one then put a comma between. Example: F1, F2 or 1,2 either way will work If can't total the field you specify; will give a message saying so and go back to question (2)

3) Field,name,format for printing?

a) field- looking for a #. ex.:
F1 or 1; F8 or 8
If used the letter "S" in question (1) then don't format it.

b) name-any name that you want printed above field on printout to designate what field contains. (ex: Sign#)

c) format - any that you would use in a "print-using" statement. For numeric fields use "##". example; for a field 8 long use "####,###.##" Make sure field is big enough. If negative # use a "-" at end of format to specify neg. on printout.

For Alpha numeric field use:
ex: 11 long - \9\ - if spaces desired before or after field:
ex: 2\9\1
number between back slashes must be 2 less than actual length be cause each "\" counts for 1

When completed all fields then <CR>

4) Field Change Subtotals?

Field # that you want a Subtotal for in your total field when Subfield changes. If more than one then seperate by commas. If none then <CR>
("*" by number will cause a top of form after subtotal)

When program has completed - Terminal will come back with "Ready"